On Trading Off Consistency and Coverage in Inductive Rule Learning

Frederik Janssen and **Johannes Fürnkranz**
TU Darmstadt
D-64289, Darmstadt, Deutschland
[janssen,juffi]@ke.informatik.tu-darmstadt.de

Abstract

Evaluation metrics for rule learning typically, in one way or another, trade off consistency and coverage. In this work, we investigate this trade-off for three different families of rule learning heuristics, all of them featuring a parameter that implements this trade-off in different guises. These heuristics are the *m*-estimate, the F-measure, and the Klösgen measures. The main goals of this work are to extend our understanding of these heuristics by visualizing their behavior via isometrics in coverage space, and to determine optimal parameter settings for them. Interestingly, even though the heuristics use quite different ways for implementing this trade-off, their optimal settings realize quite similar evaluation functions. Our empirical results on a large number of datasets demonstrate that, even though we do not use any form of pruning, the quality of the rules learned with these settings outperforms standard rule learning heuristics and approaches the performance of Ripper, a state-of-the-art rule learning system that uses extensive pruning and optimization phases.

1 Introduction

Evaluation metrics for rule learning typically, in one way or another, have to trade off consistency and coverage. On the one hand, rules should only cover a small percentage of negative examples, on the other hand, rules that cover more examples tend to be more reliable, even though they might be less precise on the training examples than alternative rules with lower coverage. An increase in coverage of a rule typically goes hand-in-hand with a decrease in consistency, and vice versa. Thus, many successful rule learning heuristics try to trade off these two aspects. For example, the well-known information gain heuristic of FOIL [Quinlan, 1996] uses a logarithmic difference between a rule and its predecessor as a measure of the increase in consistency of a rule, and multiplies this with the rule's coverage on the positives examples.

In this work, we will show that three well-known evaluation metrics, the *m*-estimate, the F-measure, and the Klösgen measures, may be interpreted as different ways for trading off consistency and coverage. Following the framework laid out in [Fürnkranz and Flach, 2005], we will first visualize their behavior in coverage space in order to demonstrate the way they implement this trade-off. Subsequently, we will report on an extensive experimental study with the goal of determining optimal values for each of these three parameters.

A more detailed description of the results of this paper, and some additional material can be found in [Janssen, 2006].

2 Inductive Rule Learning

The goal of an inductive rule learning algorithm is to automatically learn rules that allow to map the examples of the training set to their respective classes. Different algorithms implement different ways for finding individual rules, but most of them employ a *separate-and-conquer* or *covering* strategy for combining rules into a rule set.

Separate-and-conquer rule learning can be divided into two main steps: In the first one a single rule is learned from the data (the *conquer* step). Then all the (positive) examples which are covered by the learned rule are being removed from the training set (the *separate* step). The next rule is learned on the remaining examples. The two steps are repeated as long as (positive) examples are left in the training set. This ensures that every positive example is covered at least by one rule (*completeness*) and no negative example is included (*consistency*). The origin of this strategy is the AQ-Algorithm [Michalski, 1969] but it is still used in many algorithms [Fürnkranz, 1999].

3 Heuristics and the Coverage Space

In [Fürnkranz and Flach, 2005] it was suggested to visualize the behavior of rule learning heuristics by plotting their isometrics in coverage space, an un-normalized version of ROC-space. In this section, we briefly review the main concepts.

Some notational conventions

In the remainder of this paper the following notations are used:

- p and $n \equiv$ the positive/negative examples covered by the rule (local)
- P and $N \equiv$ the total amount of positive/negative examples in the training set (global)

Rule evaluation heuristics are denoted by the letter h with a subscript to differentiate between them. All heuristics depend only on the number of positive and negative examples that are covered by the rule, and are thus unable to discriminate between rules that cover the same number of positive and negative examples. So it follows that $h(R_i) \equiv h(n_i, p_i)$ holds for all rules R_i. Furthermore it is obvious that $R_1 \neq R_2 \nrightarrow h(R_1) \neq h(R_2)$.

Coverage Space
In distinction to ROC-spaces the coverage space plots the absolute number of positive examples on the *y*-axis and the absolute number of negative ones on the *x*-axis. For example the point $(0, 0)$ represents the empty theory where no example is covered at all. A good algorithm should navigate the learning process in the direction of the point $(0, P)$. It represents the optimal theory because all positive examples are covered and no negative is included. The point $(N, 0)$ represents the opposite theory, and the universal theory, covering all positive and negative examples, is located at (N, P).

Isometrics in Coverage Space
A good method to visualize the peculiarities of a heuristic is to plot their isometrics into a coverage space. A single line of an isometric connects different points (n_i, p_i) with $n_i \in N$ and $p_i \in P$ in this space. Each of these points represents a rule R_i which covers a certain amount of positive (p_i) and negative (n_i) examples. Note that two different rules R_1 and R_2 which covers the same amount of examples, receive the same evaluation value. Isometrics connects those rules $R_1, ..., R_m$ that have the same evaluation value but covering a different amount of examples. Figure 1 shows an example of a coverage space which contains isomterics with their evaluation values.[1]

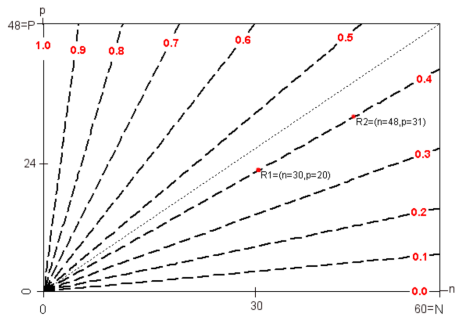


Figure 1: Isometrics in coverage space

Note that there are fewer positive than negative examples in Figure 1 which is necessary for pointing out some special differences between the heuristics (it is important that the positive and negative examples are not equally distributed). In this example the steepest line (the y-axis) represents the greatest evaluation value, which is assigned to all rules that cover some positive but no negative examples.

Based on their isometric structure, we can discern three basic types of heuristics:

- linear isometrics that are not parallel (like the one in Figure 1),
- linear ones that are parallel and
- non-linear ones.

It was shown in [Fürnkranz and Flach, 2005] that linear isometrics may be reduced to two fundamental prototypes: The first one is precision, which tries to optimize the Area under the ROC-Curve for unknown costs and the second one is a cost based optimization for known or expected costs.

[1]For visualization, one is primarily interested in the shape of the isomterics. In this case, the evaluation value is usually ommitted from the graph.

4 Overview of the Used Heuristics

A heuristic is a function that tries to find promising rules by evaluating their coverage of positive and negative examples of the training set. There are two main goals which should be taken into account if an appropriate heuristic is constructed:

- on the one hand the number of positive examples that are covered by the rule should be maximized and
- on the other hand the amount of negative examples that are covered by the rule should be minimized.

A simple way of achieving both objectives is to subtract the number of covered negatives from the covered positives. The resulting heuristic ($h_{Accuracy} = p - n$) is equivalent to accuracy, which computes the percentage of correctly classified examples in all training examples. Other heuristics employ more complex ways to reach these two objectives.

Note that $h_{Accuracy}$ may already be interpreted as a simple way of trading off coverage (represented through the maximization of *p*) and consistency (represented through the minimization of *n*). However, this trade-off is fixed and corresponds to a cost assumption (false positives and false negatives have equal costs) that does not necessarily hold in practice, and, more importantly, may not lead to a good choice of rules.

In the following, we will have a closer look at three heuristics which implement a parametrized form to trade off between coverage and consistency. In the remainder they are called the parametrized heuristics. All three heuristics measure consistency with the same metric, *Precision*, but employ different ways for measuring coverage.

In the following section we will describe these basic heuristics before we will discuss the parametrized heuristics in Section 4.2.

4.1 The basic heuristics

- Precision $h_{Precision} = \frac{p}{p+n}$

A rule is beeing evaluated with the amount of correctly classified examples (*p*) among all covered examples (*p*+*n*). This heuristics picks the steepest line in the PN-space. Its isometrics rotating around the origin as can be seen in Figure 1 which plots those of *Precision*.

- Recall $h_{Recall} = \frac{p}{P}$

This one evaluates a rule with the fraction of covered positive examples in all positive examples of the training set. This estimation is independet of the covered negative examples which results in horizontal parallel lines. The toggling line receives the highest evaluation value because the rules that are located on this one cover the most positive examples.

- Coverage $h_{Coverage} = \frac{p+n}{P+N}$

The idea of this heuristic is similiar to the concept of *Recall*, but the covered negative examples are taken into account as well. The maximum heuristic value is reached if all examples of the training set are covered. In that case the rule corresponds to the point (N, P) of the coverage space and represents the universal theory. The isometrics are lines with a slope of -1.

- WRA $h_{WRA} = \frac{p+n}{P+N} \cdot \left(\frac{p}{p+n} - \frac{P}{P+N} \right) \sim \frac{p}{P} - \frac{n}{N}$

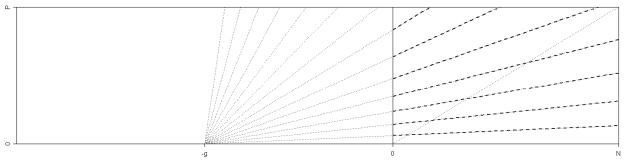


Figure 2: General behavior of the F-Measure

The basic idea of *weighted relative accuracy* (WRA) [Lavrac *et al.*, 1999] is to compute accuracy on a normalized distribution of positive and negative examples. As a result, the lines of the isometrics are now parallel to the diagonal of the coverage space instead of those of $h_{Accuracy}$ which have a slope of 1 (and are independet from the *a priori* class distribution).

WRA differs from the other two coverage-heuristics because it does not directly optimize coverage alone. In fact, like accuracy, it already implements a fixed trade-off between consistency and coverage. However, the experimental evidence of [Todorovski *et al.*, 2000] (which is consistent with our own experience) suggests that this measure has a tendency to over-generalize, i.e., that it places too strong emphasis on coverage.

4.2 The parametrized heuristics

The heuristics that we consider in this work all have a parameter that allows to gradually transform the isometrics of $h_{Precision}$ into one of the three coverage-based metrics that we discussed in the previous section. In the following, we will analzye the changes which happen during this process. If we are able to see how the preferences of the heuristic are modified, we can develop a better understanding of these heuristics and the trade-off they implement.

- F-measure $\qquad h_{F\text{-}Measure} = \frac{\left(\beta^2+1\right)\cdot h_{Precision}\cdot h_{Recall}}{\beta^2\cdot h_{Precision}+h_{Recall}}$

The F-measure [Salton and McGill, 1986] has its origin in Information Retrieval and trades off the basic heuristics $h_{Precision}$ and h_{Recall}. There are some common parametrizations which either focus on the influence of *Precision* or *Recall* or trade them off equally. If $\beta \to 0$ the isometrics correspond to those of $h_{Precision}$ as shown in Figure 2 for $g = 0$. The more the parameter is increased the more the origin of the isometrics is shifted in the direction of the negative N-axis. The observable effect is that the lines in the isometrics becomes flatter and flatter. Finally if $\beta \to \infty$ the resulting isometrics approach those of h_{Recall} which are horizontal parallel lines.

- m-estimate $\qquad h_{m\text{-}estimate} = \frac{p+m\cdot\frac{P}{P+N}}{p+n+m}$

The idea of this parametrized heuristic [Cestnik, 1990] is to presume that a rule covers m training examples *a priori*, which are assumed to be distributed according to the distribution of the examples in the training set $\frac{P}{P+N}$. There is a common parameter setting of $m = 2.0$. In this case – assuming an equal example distribution – we get the *Laplace* heuristic:

$h_{Laplace} = \frac{p+2.0\cdot\frac{1}{2}}{p+n+2.0} = \frac{p+1.0}{p+n+2.0}$

If we inspect the isometrics in relation to the pass through the different parameter settings, we observe a shift of the origin of the coverage space. Related to the situation that was described at the F-measure the origin is moved to the point $(-n_m, -p_m)$ with $p_m = m \cdot \frac{P}{P+N}$ and $n_m = m - p_m$. Here it is shifted in the direction of the negative diagonal of the coverage space as can be seen in Figure 3.

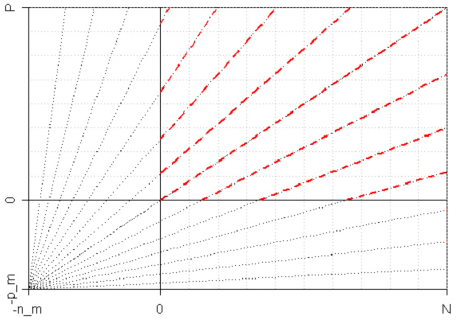


Figure 3: General behavior of the m-estimate

The more the parameter is increased the more the lines become parallel. If $m \to \infty$ the lines are parallel to the diagonal and match the isometrics of h_{WRA}. Thus, the m-estimate performs a trade-off between *Precision* and WRA.

- Klösgen $h_{Klösgen} = \left(h_{Coverage}\right)^{\omega} \cdot \left(h_{Precision} - \frac{P}{P+N}\right)$

This family of measures was first proposed by Klösgen [Klösgen, 1992] and trades off *Precision Gain* (the increase in precision compared to the default distribution $P/(P+N)$) and *Coverage*. Thus *Precision Gain*, as opposed to *Precision*, takes the *a priori* distribution into account.

Klösgen suggested the parameter settings $\omega = 0.5$ and $\omega = 1$, the parameter $\omega = 2$ was investigated by [Wrobel, 1997]. Setting $\omega = 1$ results in WRA, and $\omega = 0$ yields *Precision Gain*, which has the same isometric structure as *Precision* because they only differ by the substraction of a constant. Thus, the Klösgen measure starts with the isometrics of $h_{Precision}$ and first evolves into those of h_{WRA}, just like the m-estimate. However, the transformation takes a different route, as shown in Figure 4. Graph A shows that the lines in the area of low coverage are bent towards the diagonal of the coverage space if the parameter is increased. This indicates a preference for rules which cover few examples. The bending of the lines decreases with an increase of the parameter until they are parallel. The isometrics of picture B comply with those of h_{WRA}.

If the parameter is increased further on, the isometrics converge to $h_{Coverage}$, as shown in Figure 5. Graph A reflects the parameter setting suggested by Wrobel. Here the region of few examples is avoided because the influence of the coverage is increased. Thus the evaluation value is higher the more the lines move away from the diagonal. In picture B this effect is further strenghtend by increasing the parameter to 7.0. Additionally the rules which are evaluated better move towards the point (N, P). If one is looking at a single line in graphic B it starts with a certain amount of positive examples and no negatives. It then shows an almost linear decrease of covered positives and

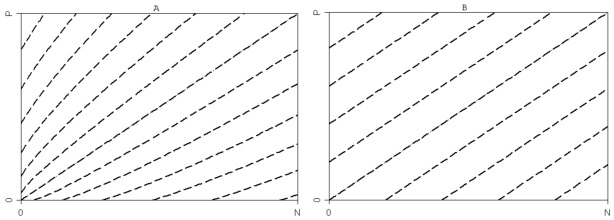


Figure 4: Klösgen-Measure for $\omega \leq 1$

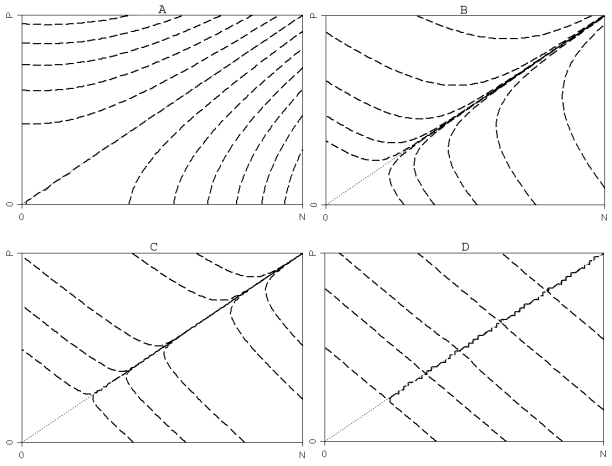


Figure 5: Klösgen-Measure for $\omega > 1$

increase of covered positives, very similar to *Coverage*. However, near the diagonal, the coverage of positive examples suddenly increases as well as those from the negative ones. This behavior is known from WRA. The influence of this heuristic is decreased more and more. This effect is visualized near the diagonal where the lines of the isometric becomes increasingly parallel (graph C in Figure 5). Finally, for $\omega \to \infty$, the influence of WRA is abjured and the isometrics match those of $h_{Coverage}$.

Another interesting variation of the Klösgen measure is to divide $h_{Coverage}$ by $1 - h_{Coverage}$ instead of raising it to the ω-th power. This turns out to be equivalent to the heuristic *Correlation* ($h_{Corr} = \frac{p \cdot (N-n) - n \cdot (P-p)}{\sqrt{P \cdot N \cdot (p+n) \cdot (P-p+N-n)}}$) as has been shown in [Klösgen, 1992].

5 Experimental setup

The primary goal of our experimental work was to determine settings for the parametrized heuristics that are optimal in the sense that they will result in the best overall performance on a wide variety of datasets. Clearly, the optimal setting for individual datasets may vary.

There are some important points that we have to keep in mind when performing the search for the best parameter. First, a large amount of datasets ought to be employed. This is necessary to be independent of special characteristics of them. Second, the parameters should be searched on some datasets and then be tested on an independent set of datasets. This step is important to assure that the obtained parameters are universally valid.

5.1 The datasets

We have used 27 datasets of the UCI-Repository [Newman *et al.*, 1998] for the search of the parameters. It was not important how the datasets were constituted (the number of attributes, classes and examples are indifferent). We chose the following ones because the quantity of examples varies from 24 to 8124, the number of attributes moves between 3 and 69 and finally the lowest amount of classes is 2 and the highest 24:

> *anneal*, *audiology*, *breast-cancer*, *cleveland-heart-disease*, *contact-lenses*, *credit*, *glass2*, *glass*, *hepatitis*, *horse-colic*, *hypothyroid*, *iris*, *krkp*, *labor*, *lymphography*, *monk1*, *monk2*, *monk3*, *mushroom*, *sick-euthyroid*, *soybean*, *tic-tac-toe*, *titanic*, *vote-1*, *vote*, *vowel*, *wine*.

Then the obtained parameters were tested on 30 different datasets that were also taken from the UCI-Repository. Here similiar contraints were valid. In this datasets the number of examples diversifies from 57 to 2310, the count of attributes goes from 4 to 60 and the number of classes is between 2 and 22. The sets were:

> *auto-mpg*, *autos*, *balance-scale*, *balloons*, *breast-w*, *breast-w-d*, *bridges2*, *colic*, *colic.ORIG*, *credit-a*, *credit-g*, *diabetes*, *echocardiogram*, *flag*, *hayes-roth*, *heart-c*, *heart-h*, *heart-statlog*, *house-votes-84*, *ionosphere*, *labor-d*, *lymph*, *machine*, *primary-tumor*, *promoters*, *segment*, *solar-flare*, *sonar*, *vehicle*, *zoo*.

All given accuracies are calculated with a *1x10-stratified Cross Validation* implemented in *weka* [Witten and Frank, 2005].

5.2 The rule learner

We have used a separate-and-conquer rule-learner that is implemented within the SECO-Framework [Fürnkranz, 1999; Thiel, 2005], which is a modular architecture for rule learning that is under development in group. The framework defines a generic separate-and-conquer rule learner that allows to configure specific variations by specifying appropriate modules. In our study, we only varied the heuristics and kept all other options simple and stable. It is not a fundamental point which rule-learner was used because we aim more at an empiric study about different rule learning heuristics than at experiments about various methods of rule learning. The search strategy was chosen to be *Top-Down Hill-Climbing* and no special stopping criterion was used to avoid overfitting because we wanted to solely focus on the heuristics' abilities to evaluate the quality of a rule.

5.3 The evaluation methods

As we have a large number of different individual results, a key issue is how to determine which parameter performed best on average. We have experimented with several choices.

Our primary method was the *Macro-Averaged-Accuracy* of one parametrization of a parametrized heuristic on all of the datasets. Assume that there are m datasets overall. The correctly classified examples of dataset *i* are denoted by $corr_i$ and the total amount of examples of dataset *i* is called $total_i$.

Defintion 5.1 (Macro-Averaged-Accuracy) *The* Macro-Averaged-Accuracy *is computed in two steps: First the accuracy of a heuristic on a single dataset is calculated by dividing the correctly classified by the total number of examples in the corresponding set. Then the accuracy of all datasets is averaged.*

$$Av\_Acc_{macro} = \frac{\sum_{i=1}^{m} \frac{corr_i}{total_i}}{m}$$

However, there are other sensible choices for combining individual results. For examples, *Macro-Averaged-Accuracy* gives the same weight to all datasets. Alternatively, one could assign the same weight to each misclassified example, which results in *Micro-Averaged-Accuracy*. This method assigns a higher weight to datasets with many examples and those with few examples get a minor weight.

Defintion 5.2 (Micro-Averaged-Accuracy) Micro-Averaged Accuracy *is computed by dividing the number of correctly classified examples on* all *different datasets by the total number of examples in all datasets.*

$$Av\_Acc_{micro} = \frac{\sum_{i=1}^{m} corr_i}{\sum_{i=1}^{m} total_i}$$

As there are large differences in the variances of the accuracies of the individual datasets, one could also focus only on the *Ranking* of the heuristics and neglect the magnitude of the accuracies on different datasets. For example, if one heuristic achieves 90.25 % and another one gets 90.23 % on the dataset this difference is not really taken into account when calculating the *Macro-Averaged-Accuracy* on a great number of datasets. In this case the Ranking method provides a better separation because the first heuristic gets rank number 1 and the second rank number 2.[2] Small variations will cancel out over multiple datasets, but if there is a constant small advantage of one heuristic over the other it may be better observed on a combined ranking than on on an averaged accuracy value.

The rankings of the heuristics are combined by adding up their individual ranks.

Defintion 5.3 (Average Rank) *The* Average Rank *is the average of the individual ranks r_i on each dataset.*

$$Av\_Rank = \frac{\sum_{i=1}^{m} r_i}{m}$$

During the search for the optimal setting we selected a large set of interesting parameter settings. All of these parameters are taken as individual heuristics described by their name and the corresponding parameter which leads to a total of 45 parametrized heuristics. For example the general Klösgen measure which are initialized with a parameter of 2.0 is called *Klösgen2.0*. As a result of the evaluation, we have created two tables containing all the heuristics with their *Macro-*, their *Micro-Averaged-Accuracy*, and their *rank*. In addition, we also measured the total *number of rules and conditions*. The first table is produced on the results obtained on the 27 sets on which the parameters have been searched and the second one corresponds to the outcomes on the 30 sets used to evaluate the heuristics (cf. Section 5.1). The correlation of the two tables is an indicator for the universality of the determined parameters. The higher the correlation value, the more reliably will the parameters work on arbitrary datasets. The comparison is made by a correlation value calculated with the *Spearman Rank Correlation*.

Defintion 5.4 (Spearman Rank Correlation) *Given two (averaged and rounded) rankings r_i and r'_i for the heuristics $h_i, i = 1 \ldots m$, the* Spearman Rank Correlation *is defined as*

$$\rho = 1 - \frac{6 \cdot \sum_{i=1}^{m} (r_i - r'_i)^2}{m \cdot (m^2 - 1)}$$

[2]Ties in the ranking are handled by assigning the average rank to all tied accuracies. For examples, if the heuristics on the ranks 5–8 all have equal accuracies, they all receive the rank 6.5 on this dataset.

It computes a correlation value between -1 (which stands for a perfect negative correlation) and 1 (which represents the perfect positive correlation). A result of 0 means no correlation at all. There are some advantages of using this method:

- it is robust against anomalies and
- it is praticable for variables whose relation is non-linear.

6 Searching for the optimal parameter

This section describes our method for searching for the optimal parameter setting. First, we tested a wide range of intuitively appealing parameter settings to get an idea of the general behavior and the differences of the three parametrized heuristics. The promising parameters were restricted further on. Our expectation was to have a general behavior like the one shown in Figure 6.

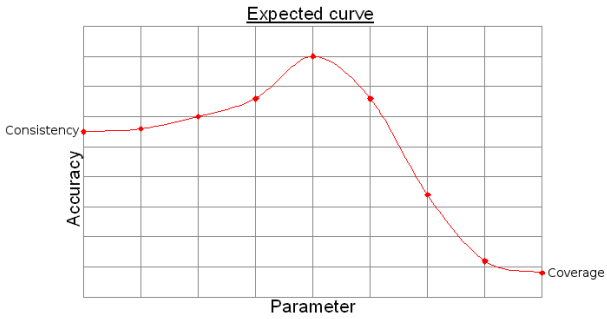


Figure 6: Expected curve

The start point, where the parameter is close to 0, represents the accuracy reached by *Precision*. Then the correctness should raise with an increase of the parameter until the optimal one is found (where the *Macro-Averaged-Accuracy* is the highest). If the parameter is increased further on, the accuracy should decrease again and should converge towards the value achieved by the used coverage heuristic. In Figure 6 the accuracy of *Precision* (which represents *Consistency*) is higher than those of the related coverage heuristic. As we will see, this holds for the Klösgen measures and the F-measure. At the m-estimate the situation is reversed because the coverage heuristic WRA achieves a higher accuracy than *Precision*.

6.1 The search strategy

As mentioned above, we first used a fixed set of parameters in order to identify the basic regions where the optimal parameter can be. After this first test, the general search algorithm is started.

There are some constraints which underlay the search method. First it should be clear that the only way to find the optimal parameter is to perform an exhaustive search through the space of all possible parameter settings. Due to limited computational power it is of course not possible to use this kind of search in practice. Another way of searching the above-mentioned space is to simply test different parameters in a certain interval and analyze promising ones further more. A good method to do this is to use nested intervals. We start with the best parameter found so far, and

Algorithm 1 The search algorithm

```
PROCEDURE SearchBestParameter (a, b, i, currHeur, dataSets)
{
   p_best = 0
   acc_best = 0
   acc_former = acc_best
   t = 0.001
   # create a list (params) containing the parameters to
   # search
   params = createList(a, b, i)
   # find the best parameter in this list
   p_best = getBestParam(currHeur, params, dataSets)
   # get the highest accuracy
   acc_best = getHighestAcc(currHeur, params, dataSets)
   # if no significant improvement is yielded return the
   # best parameter and break
   IF (acc_best − acc_former < t)
   {
     RETURN (p_best)
     BREAK
   }
   # call the procedure recursively with the new borders
   # and the new increment
   SearchBestParameter(p_best − i/2, p_best + i/2, i/10, currHeur,
   dataSets)
}
```

divide the previous increment by a predefined value. Next, a certain interval around this best parameter is searched for a better value. If one is found, it is refined using the same method as above. So basically a certain value is selected out of an interval and a new interval is created around the value. Then the next interval is selected out of the previous one. If the length of the intervall is converging towards 0, a real number is yielded which lies in every intervall. An example search is shown in Table 1.

There are several constraints of setting the parameters of the search. For example, the farther the lower border *a* and the upper border *b* of the related interval are away from the best parameter p_{best}, the higher the probability is that the global optimal parameter will be found. Due to the restricted calculation power the constraints are defined as follows (*i* stands for the increment):

$$a = p_{best} - \frac{i}{2},\ b = p_{best} + \frac{i}{2} \text{ and } i = \frac{i}{10}$$

Additionally a threshold *t* for minimum accuracy differences has to be initialized. Suitable values could be derived from significance tests, but we simply set this value to 0.001. A schematic description of the search algorithm is given in pseudo code at Algorithm 1.

There are some problems resulting out of the proposed method:

- there is a possibility to simply miss the best parameter due to the fact that the global best parameter may lie under or above the borders (if the best one so far is 1 for example, the interval that would be searched is $[0.5, 1.5]$; if the global optimum is 0.4, it would not be detected)
- there is only one possible optimum that is closer examined (the global optimum could hide between two apparently lower values)

The latter can be addressed by keeping a list of candidate parameters that all be refined and from which the best one is selected. One has to define how many candidates should be maintained. Therefore it is necessary to introduce a threshold that discriminates between a normal and a candidate parameter. It is not trivial to determine such a threshold. Due to this the number of candidate parameters is limited to 3 (all experiments confirmed that this is sufficient). The first problem remains unresolved. Because of complexity issues the borders have to be adjusted as proposed. The focus of this work is not to find the global optimum definitely.[3] Instead, we aim at identifying interesting intervals of the 3 parametrized heuristics. If we can find the region that is likely to contain the best parameter, independent from the datasets, this would already be a sufficiently interesting result.

Table 1: A sample parameter search

Run	set which has to be searched	increment	best parameter	Accuracy
1	$\{0.1, \ldots, 1.0\}$	0.1	0.4	84.5658
2	$\{0.35, \ldots, 0.45\}$	0.01	0.42	84.6852
3	$\{0.415, \ldots, 0.425\}$	0.001	0.418	84.7015
4	$\{0.4175, \ldots, 0.4185\}$	0.0001	0.4176	84.7045
5	$\{0.41755, \ldots, 0.41765\}$	0.00001	0.4176	84.7045

6.2 Optimal parameters for the three heuristics

In this section we focus on the results of the search and describe the different parameters we have found for the three heuristics. In addition, we introduce graphs in which we plot curves that show interpoloated accuracy values over various parameter settings. These curves illustrate the behavior of the different parameter settings.

Klösgen measures

Figure 7 (a) shows the results for the Klösgen measures. The curve corresponds to our expectations (cf. Figure 6). In the region from 0.1 to 0.4 the accuracy increases continuously until it reaches a global optimum at 0.4323, which achieves an accuracy of $84.9909\,\%$. After the second run of the search algorithm no better candidate parameters were found. The accuracy decreases again with parametrizations greater than 0.6. The parameter setting of 1.0 represents WRA. Larger values are not shown, as they turned out to further decrease the accuracy. As illustrated in Figure 4, the shown interval $[0, 1]$ describes the trade-off between *Precision* and WRA. So one can say that the trade-off between WRA and *Coverage*, which is obained for values of $\omega > 1$, does not reach a sufficient accuracy and can therefore be ignored.

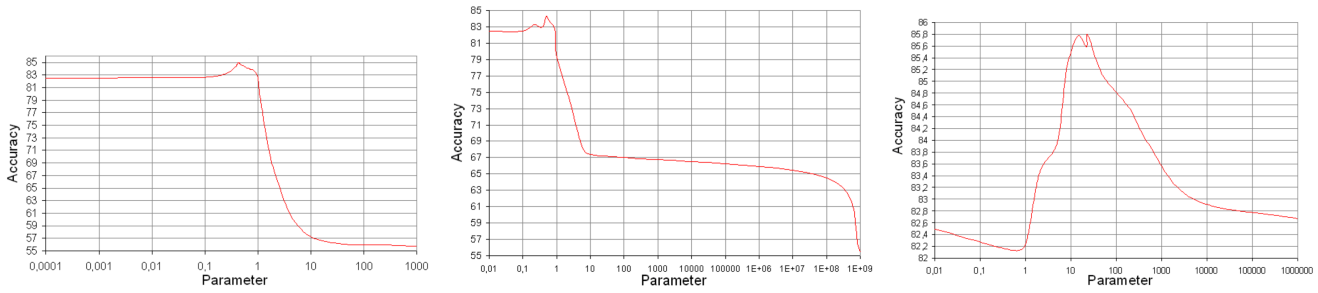
F-measure

For this heuristic the same interval as with the Klösgen measures is of special interest (Figure 7 (b)). Already after the first run the parameter 0.5 got the highest accuracy of $82.2904\,\%$. A better one could not be found during the following runs of the algorithm. After the second pass two other candidate parameters, namely 0.493 with $84.1025\,\%$ and 0.509 with $84.2606\,\%$ were found. But both of them could not be refined to achieve a higher accuracy and were therefore ignored. The main difference between the Klösgen measures and the F-measure was, that for the latter, the accurary has a steep descent at a very high parametrization of $1 \cdot E^9$. At this point it reaches the same value as the Klösgen measures (about $55\,\%$).

m-estimate

The behavior of this heuristic differs from the other two parametrized heuristics in several ways. For example, we even noticed a decrease for low parameter settings (Figure 7 (c)). The main problem is that the first run exhibited no clear tendencies. So the region in which the best parameter should be could not be restricted, and we had to search a larger interval. In Figure 8 we zoom into the range

[3] The optimal parameter will change anyhow if it is searched on different datasets.



(a) Klösgen-measures (b) F-measure (c) m-estimate

Figure 7: Accuracy over parameter values for the three parametrized heuristics

$[1, 100]$ to give a better impression of the heuristic's behavior in that critical region. The figure also shows many variations, which complicated the identification of an optimal parameter range. A significant deterioration in the accuracy cannot be detected before a value of 90 where it falls permately below $82.1\,\%$.

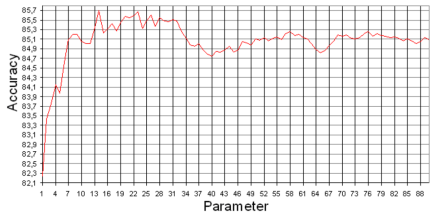


Figure 8: The curve of the m-estimate for a big interval

Due to this, the interval $[0, 35]$ had to be searched with an increment of 1 because all parameters greater than 35 got accuracies under $85.3\,\%$ and we had to restrict the area of interest. After this first run there were 3 candidate parameters, from which 14 achieves the greatest accuracy. After a second run, 23.5 became optimal, which illustrates that it was necessary to maintain a list of candidate parameters. After a sufficient amount of runs we found the opimal parameter at 22.466. The achieved accuracy of 85.8003 was the best value of all heuristics.

6.3 Behavior of the optimal heuristics

In this section, we compare the parameters which have been found for the three heuristics. The best heuristic was the m-estimate. The next one was the generalized Klösgen measures which was approximately $1\,\%$ worse, followed by the F-measure whose optimal value lagged about $0.7\,\%$ behind the generalized Klösgen measures. It is interesting to look at the isometrics of the best parameter settings of the heuristics. Interestingly, the optimal values of the m-estimate and the Klösgen measures implement a very similar heuristic, as can be seen in subfigures (b) and (c) of Figure 9. Minor differences are detectible in the low coverage region near the origin, where the isometrics of the Klösgen measures are slightly bended.

The F-measure produces somewhat different isometrics, which mostly results from its bias towards parallel lines near the N-axis, because the origin of the isometrics can only move along this axis. Therefore it can never reach an isometric structure similiar to this of the other two measures.

6.4 Validity of the results

In order to make sure that we do not overfit the datasets that were used for this study, we compared the rankings of 15 different parametrizations per heuristic on the original datasets with their rankings on new datasets, which were not used for finding the optimal values. We also added some standard heuristics (*Correlation*, *WRA*, *Precision*, *Laplace* and *Accuracy*), as well as JRIP, WEKA's implementation of RIPPER [Cohen, 1995], which, in contrast to our algorithms, uses sophisticated pruning mechanisms. In total, 52 heuristics were compared.

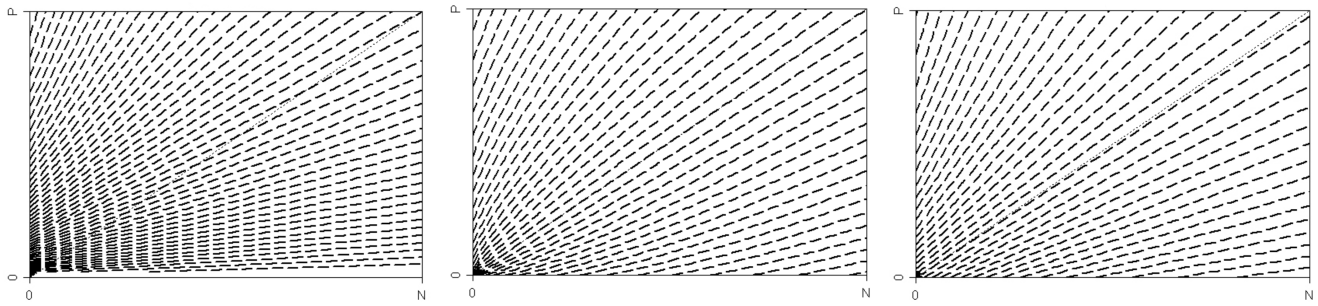
The results for the original datasets are summarized in Table 2 and for the test sets in Table 3. The numbers in braces describes the rank of each heuristic according to the measure of the respective column. The correlation value that describes the similarity between the two tables was 0.92 for Macro-, 0.91 for Micro-Averaged-Accuracy, 0.99 for the number of conditions and 0.99 for the number of rules which is not displayed in the tables. This is a very

Table 2: Different evaluations

	average accuracy		average	
Heuristic	Macro	Micro	Rank	# conditions
m-estimate22.466	85.80 (1)	93.87 (2)	16.13 (2)	36.81
Klösgen0.4323	84.99 (7)	93.62 (7)	18.69 (7)	46.89
JRip	84.45 (11)	93.80 (4)	17.37 (5)	16.93
F-measure0.5	84.29 (12)	92.94 (14)	19.07 (8)	40.78
JRip-P	83.88 (17)	93.55 (9)	21.93 (13)	45.52
Correlation	83.66 (19)	92.39 (24)	25.15 (25)	37.11
WRA	82.71 (29)	90.43 (37)	28.26 (35)	14.41
Precision	82.50 (33)	92.21 (28)	27.89 (31)	99.93
Laplace	82.28 (34)	92.26 (27)	27.30 (30)	91.04
Accuracy	82.28 (35)	91.31 (33)	28.19 (34)	84.07

Table 3: Different evaluations on the "Test Set"

	average accuracy		average	
Heuristic	Macro	Micro	Rank	# conditions
JRip	78.98 (1)	82.42 (1)	16.60 (1)	12.20
m-estimate22.466	78.68 (2)	81.72 (3)	17.97 (3)	47.27
JRip-P	78.50 (5)	82.04 (2)	18.47 (5)	49.80
Klösgen0.4323	78.49 (6)	81.33 (14)	19.87 (12)	62.67
F-measure0.5	78.14 (12)	81.52 (9)	18.27 (4)	52.43
Correlation	77.57 (22)	80.91 (21)	24.70 (26)	47.50
Laplace	76.89 (28)	79.76 (30)	26.27 (31)	118.83
Precision	76.22 (33)	79.53 (35)	29.80 (40)	129.17
WRA	75.80 (37)	79.35 (37)	27.03 (34)	12.13
Accuracy	75.60 (41)	78.47 (39)	31.23 (42)	104.77



(a) F-measure (b) Klösgen (c) m-Estimate

Figure 9: Isometrics of the best parameter settings

high correlation, which makes us confident that the found parameters will also work well on new datasets.

7 Conclusions

In this work, we investigated three different ways for trading of consistency and coverage for rule learners, in the form of three different parametrized heuristics. For each heuristic, we determined an optimal parameter, which proved to be quite stable over multiple domains. The best trade-off was achieved for the m-estimate, but the other heuristics produced quite similar behavior, which we confirmed by visualizing their isometrics in coverage space. While the exact value for this trade-off is certainly not that important, our experiments provide evidence that the optimal parameters are located in the interval $[0.3, 0.5]$ for both the Klösgen measures and the F-measure, and $[13, 27]$ for the m-estimate.

As further work we could examine other evaluation methods to find the optimal parameters. Another promising way is to re-adjust the trade-off every time a rule is learned and the examples are removed from the training set. This approach is located in the domain of Meta-Learning. Finally, we intend to look at different trade-offs between consistency and coverage, most notably to a parametrized cost metric. For these, the isometrics are always parallel lines, so that the behavior of the optimal value will necessarily be different from those studied here.

Acknowledgements

Part of this research was supported by the *German Science Foundation (DFG)* under grant no. FU 580/2-1.

References

[Cestnik, 1990] Bojan Cestnik. Estimating probabilities: A crucial task in Machine Learning. In L. Aiello, editor, *Proceedings of the 9th European Conference on Artificial Intelligence (ECAI-90)*, pages 147–150, Stockholm, Sweden, 1990. Pitman.

[Cohen, 1995] William W. Cohen. Fast Effective Rule Induction. In Armand Prieditis and Stuart Russell, editors, *Proceedings of the 12th International Conference on Machine Learning*, pages 115–123, Tahoe City, CA, July 9–12, 1995. Morgan Kaufmann.

[Fürnkranz and Flach, 2005] Johannes Fürnkranz and Peter A. Flach. ROC 'n' Rule Learning - Towards a Better Understanding of Covering Algorithms. *Machine Learning*, 58(1):39–77, January 2005.

[Fürnkranz, 1999] Johannes Fürnkranz. Separate-and-Conquer Rule Learning. *Artificial Intelligence Review*, 13(1):3–54, February 1999.

[Janssen, 2006] Frederik Janssen. Eine Untersuchung des Trade-Offs von Precision und Coverage bei Regel-Lern-Heuristiken, July 2006.

[Klösgen, 1992] Willi Klösgen. Problems for Knowledge Discovery in Databases and their Treatment in the Statistics Interpreter Explora. *International Journal of Intelligent Systems*, 7:649–673, 1992.

[Lavrac *et al.*, 1999] Nada Lavrac, Peter A. Flach, and Blaz Zupan. Rule evaluation measures: A unifying view. In *ILP*, pages 174–185, 1999.

[Michalski, 1969] Ryszard S. Michalski. On the Quasi-Minimal Solution of the Covering Problem. In *Proceedings of the 5th International Symposium on Information Processing (FCIP-69)*, volume A3 (Switching Circuits), pages 125–128, Bled, Yugoslavia, 1969.

[Newman *et al.*, 1998] D.J. Newman, C.L. Blake, S. Hettich, and C.J. Merz. UCI Repository of Machine Learning databases, 1998.

[Quinlan, 1996] J.R. Quinlan. Learning First-Order Definitions of Functions. *Journal of Artificial Intelligence Research*, 5:139–161, 1996.

[Salton and McGill, 1986] Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986.

[Thiel, 2005] Matthias Thiel. Separate and Conquer Framework und disjunktive Regeln, 2005.

[Todorovski *et al.*, 2000] Ljupco Todorovski, Peter Flach, and Nada Lavrac. Predictive performance of weighted relative accuracy. In Djamel A. Zighed, Jan Komorowski, and Jan Zytkow, editors, *4th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD2000)*, pages 255–264. Springer-Verlag, September 2000.

[Witten and Frank, 2005] Ian H. Witten and Eibe Frank. *Data Mining — Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann Publishers, 2nd edition, 2005.

[Wrobel, 1997] Stefan Wrobel. An Algorithm for Multi-relational discovery of Subgroups. In Jan Komorowski and Jan Zytkow, editors, *Proc. First European Symposion on Principles of Data Mining and Knowledge Discovery (PKDD-97)*, pages 78–87, Berlin, 1997. Springer Verlag.